

ブラックボックステスト偏重を超えて

ー グレーボックステストから考える、これからのテスト設計 ー

田嶋 享平（株式会社iTTest）



代表取締役社長 上島健二

所在地 Access

本社

〒532-0011
大阪市淀川区西中島5丁目8
番3号
新大阪サンアールビル北館
605号



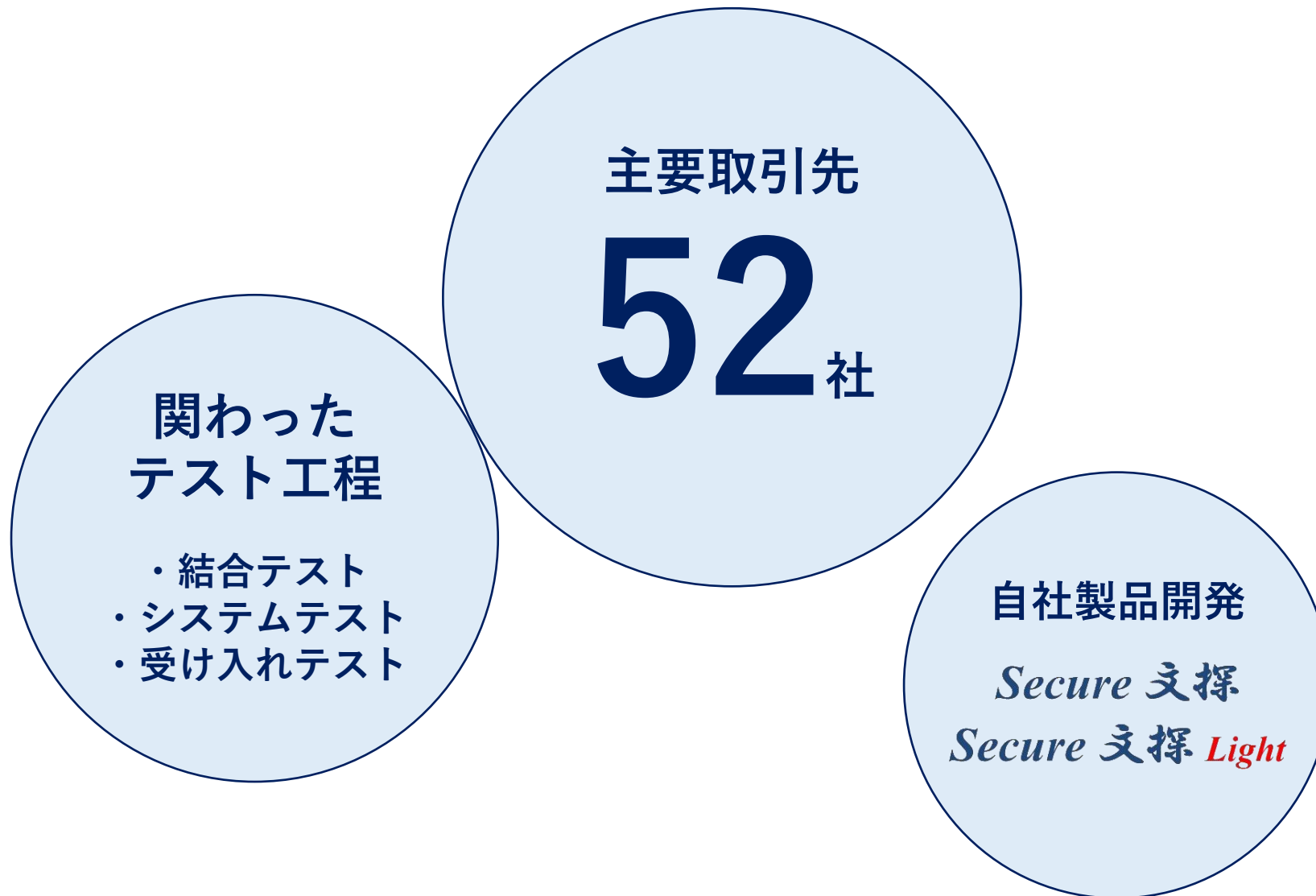
博多オフィス

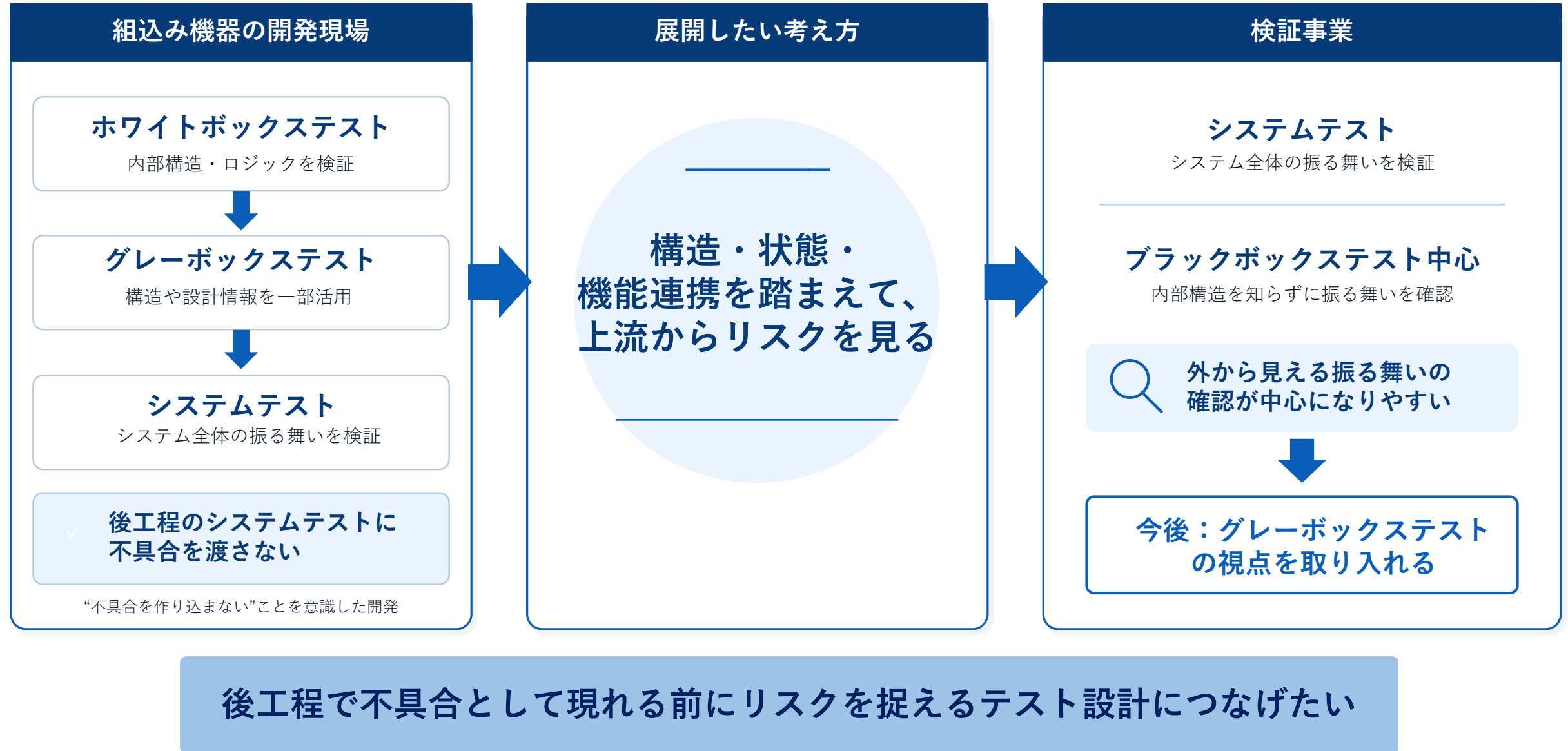
〒812-0011
福岡市博多区博多駅前3-4-
25
アクロスキューブ博多駅前



沿革 History

- 2003年 ● 資本金300万円にて有限会社マイクロテスト設立
神戸ポートアイランド内KIOにてスタート(1月)
- 2004年 ● 本社大阪市淀川区に移転
福井システムリサーチと合併
- 2005年 ● 資本金1,000万円に増資し、
株式会社 iTestへ組織および社名変更
- 2007年 ● 資本金2,305万円に増資
- 2008年 ● 資本金2,540万円に増資
- 2013年 ● 株式会社iTest 10周年
- 2018年 ● 株式会社iTest 15周年
- 2023年 ● 株式会社iTest 20周年





田嶋 享平（テストエンジニア）

- 2007年入社。
- 複合機などの民生品からFA機器まで。
- 自社製品開発では、営業支援、製品導入、アフターフォローにも携わる。



若手

仕様書の記述をそのまま
テストケース化してしまう



ベテラン

仕様書に書かれた振る舞いの
確認が中心

十分にテストしたつもりでも、後工程で不具合が見つかる。



共通点：仕様に書かれた「見える振る舞い」に寄りすぎている。

課題

外部仕様に基づくブラックボックステストはユーザー視点に優れる一方で、複雑なシステムにおいては以下のリスクが顕在化。

- 内部状態の遷移に伴うレアケースを見落としやすい
- 非同期処理・機能間連携のリスクが見えにくい
- 網羅性の根拠を説明しにくい

「仕様を確認した」だけでは、内部状態や処理に潜むリスクを捉えきれない。

グレーボックステスト（GB）の視点で上流からリスクを
早期に捉える考え方を紹介します。

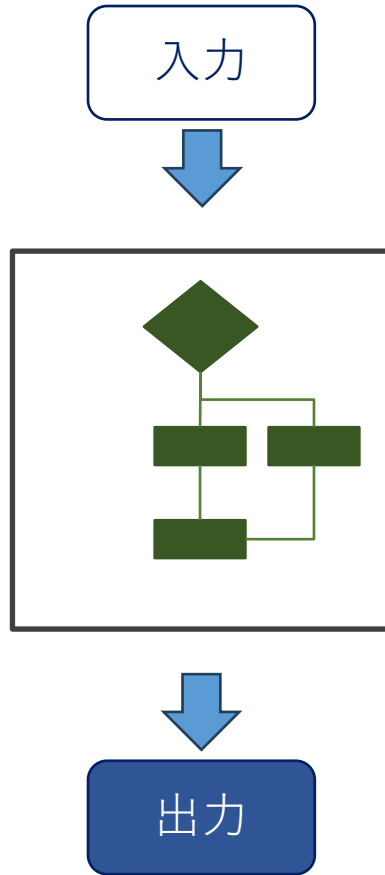
グレーボックス テスト

構造・状態・連携を
観点化し、不具合を早期に発見

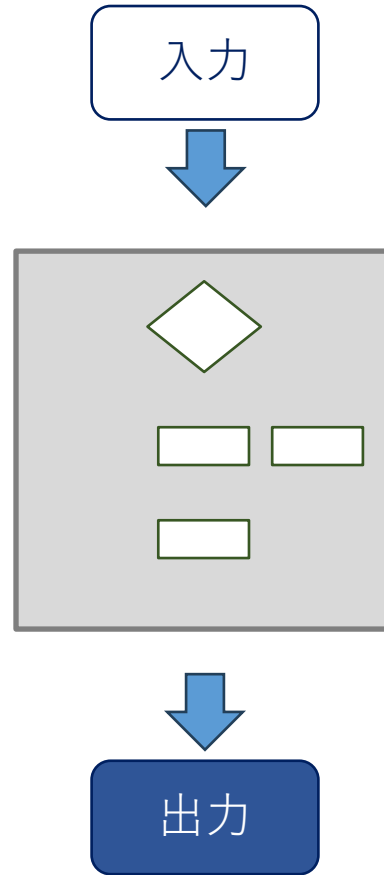
ブラックボックス テスト

ユーザー視点・仕様確認

ホワイトボックステスト

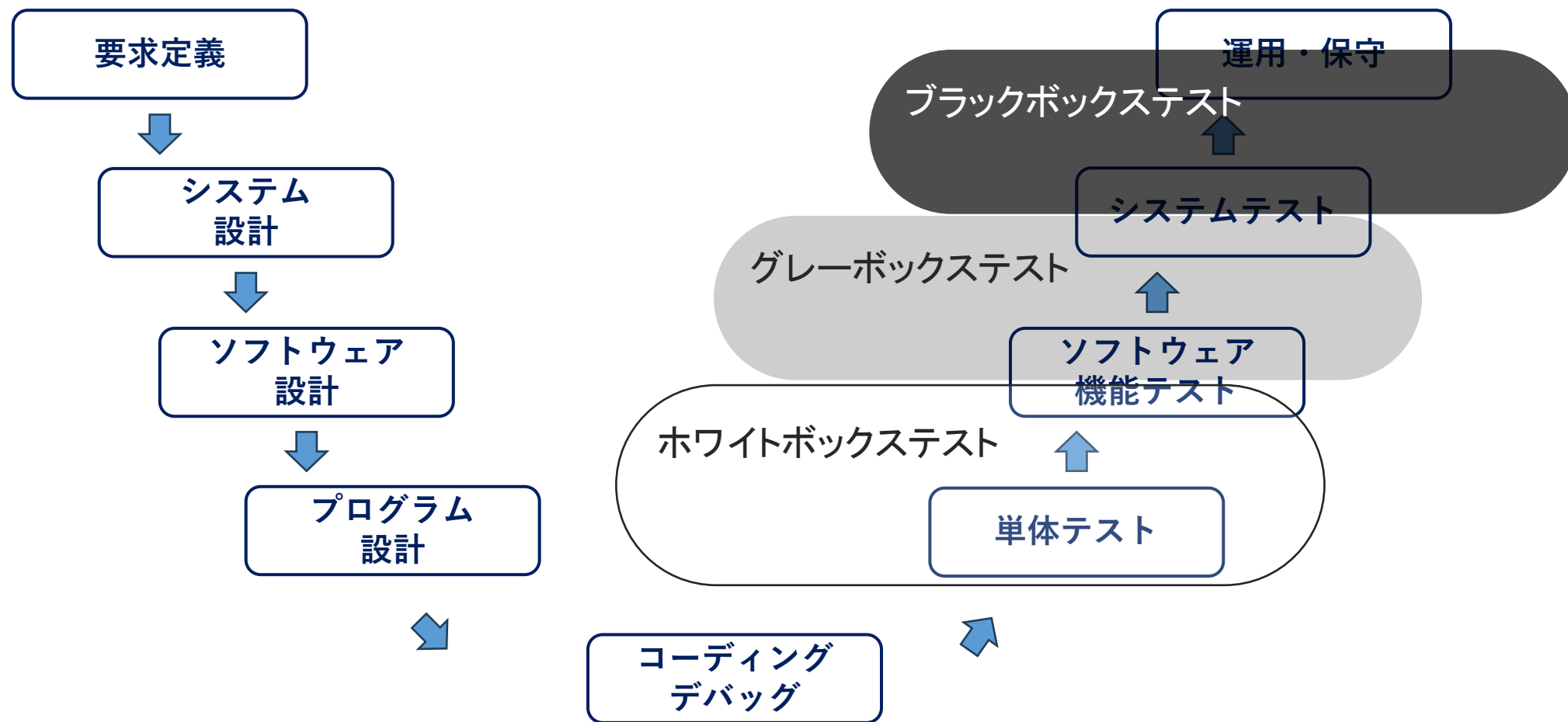


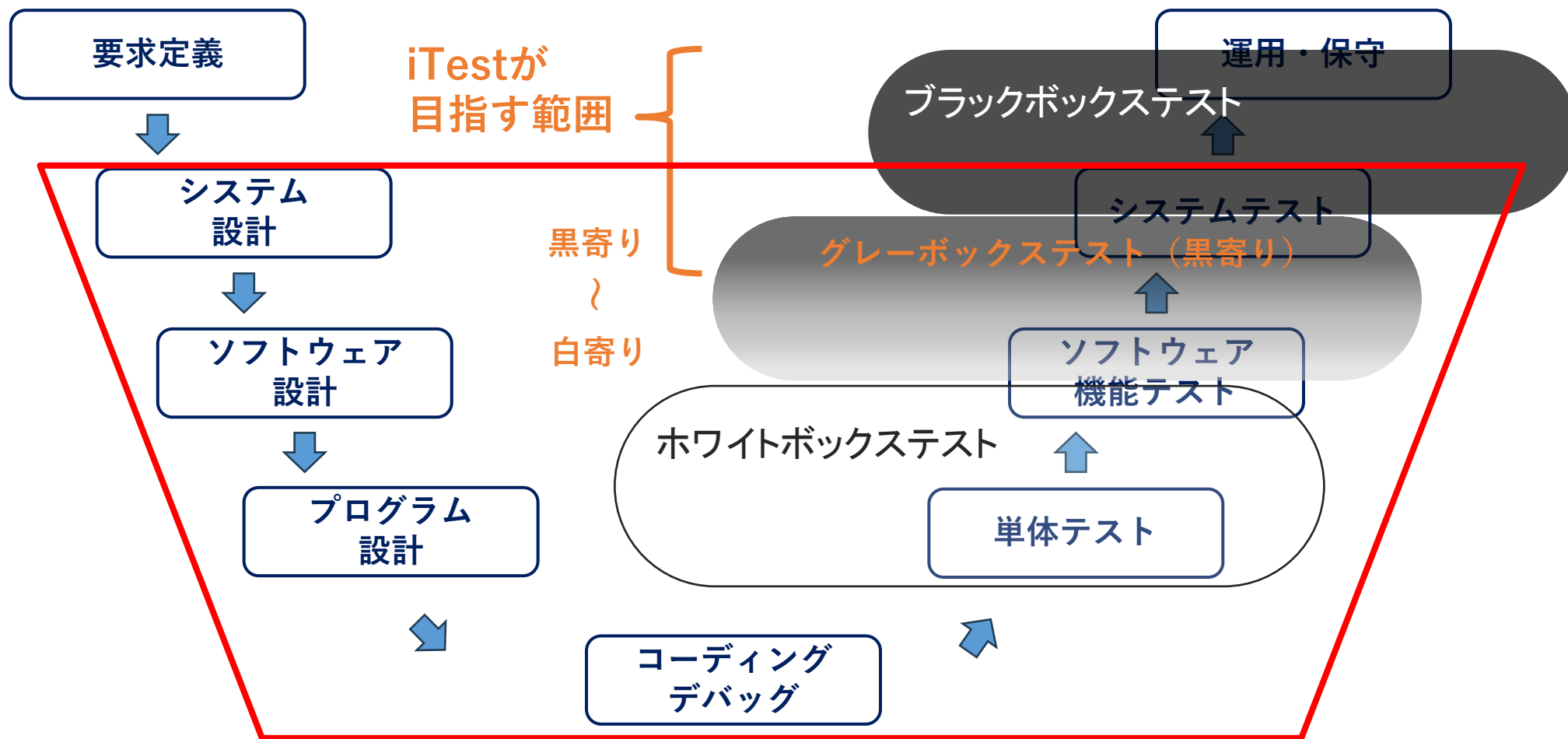
グレーボックステスト



ブラックボックステスト







	ブラックボックステスト	グレーボックステスト
対象	外から見える振る舞い	構造、状態、機能連携
確認内容	ユーザー操作、入出力、 仕様どおりの動作	状態遷移、処理タイミング、機能連携、 異常時制御
適用場面	システムテスト、受け入れテスト	結合テスト～システムテスト
注意点	内部状態や連携のリスクは見えにくい	設計情報の理解や開発側との連携が必要

狙いはBBとGBそれぞれの特徴を活かし、
不具合につながるリスクを早期に捉え、検出力を高めること

テスト開発プロセス

※「テスト設計コンテストOpenチュートリアル2026」より

1

要求分析

品質リスクを識別

2

テストアーキテクチャ設計

構造・連携を可視化

3

詳細設計

GB観点をケース化

4

実行・評価

設計した観点を確認

GBをテスト設計に取り入れるときの考え方

内部をすべて見るのではなく、BBでは見えにくいリスクを上流から観点化する

1 BBで見える振る舞いを確認

外部仕様・ユーザー操作・入出力を確認する

2 必要な範囲で構造・状態を理解

状態遷移・通信・連携・異常時の復旧を見る

3 後工程に逃しやすいリスクを観点化

「この状態で割り込みが入ったら？」 「通信異常が続いたら？」 を
テスト観点に落とす

対象：データロガーの印刷制御機能

- 典型的な組込みシステムをモデル化
- テスト設計の経験が浅いメンバーが担当
- BB設計とGB設計の差分を比較
- 追加観点のリスクを簡易評価

※障害件数ではなく、テスト観点の補強効果を測定するための社内実験

Before: 外部仕様書のみによるBB設計



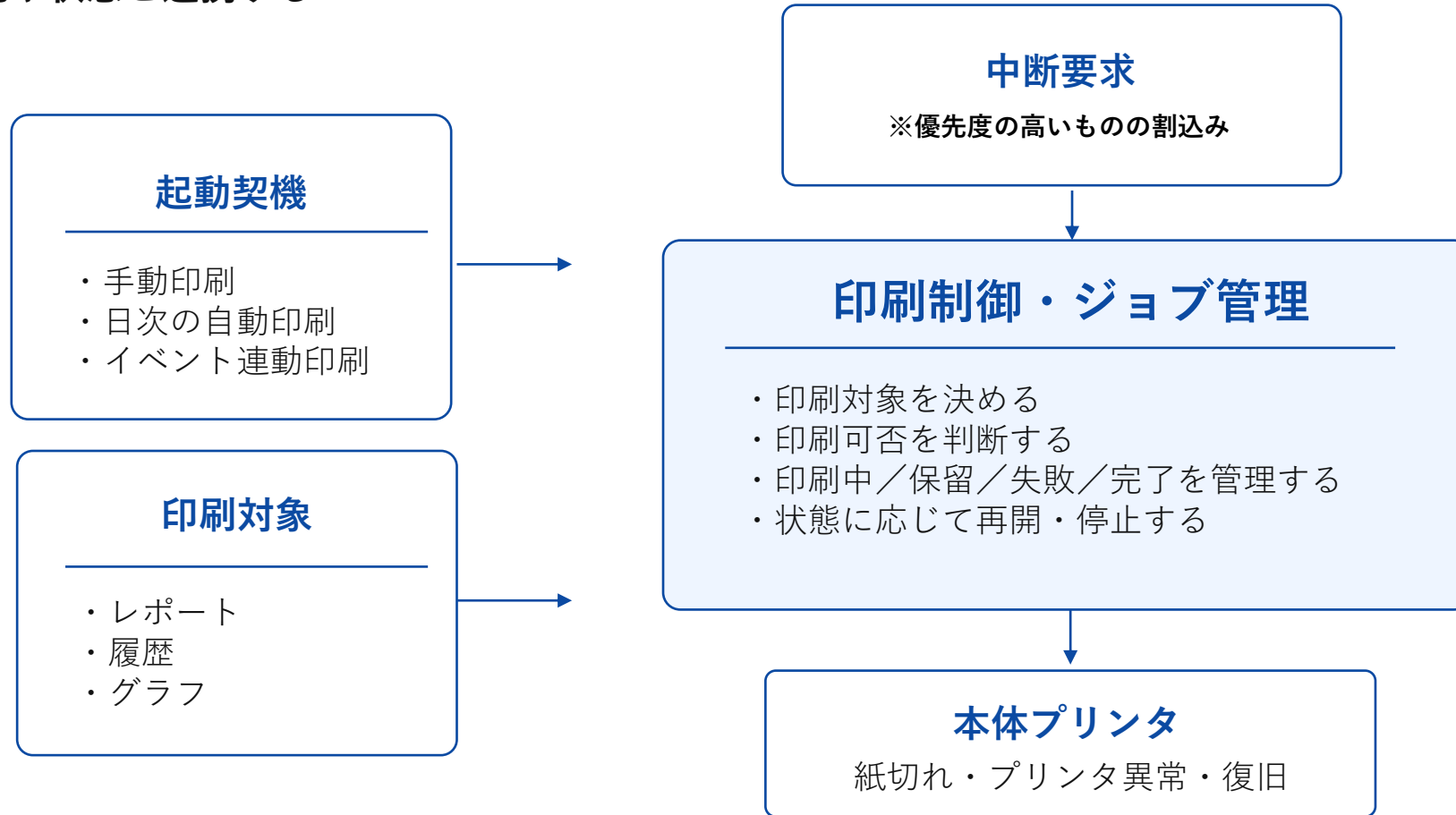
After: 構成図・フロー図を用いたGB設計



※架空のイメージです

データロガーの印刷機能

- ・優先制御や異常時制御がある
- ・複数機能や状態と連携する



観点の補完率

6.2%

※BB成果物65件⇒69件

追加観点 (GB)

4件

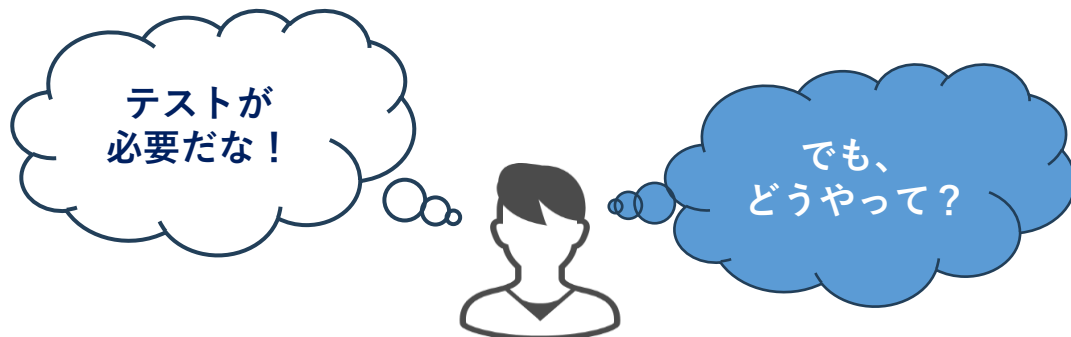
うち リスク大：2件

※リスク評価は思考実験による簡易評価。

リスク評価	件数	想定される事象例
大：優先項目	2	連携機能の処理妨害、または状態遷移不良による印刷実行不能
中：追加推奨	2	電源投入時の状態残存による手動印刷受付不可、不要な表示

不具合につながり得るリスクを早く捉えた

- GBでリスク大の観点を拾えた。では、どうやって早く確認するか。



問題点①：実機が用意できない
⇒実機や相手となる周辺機器がなくてもテストが可能

問題点②：通信プロトコル依存などテストで再現が困難
⇒異常条件を意図的に作り出して再現・観測する



GBで捉えた高リスク観点を、
実機で検証する前にシミュレー
ターで再現・確認する。

デモ：通信異常を発生させ、リトライ仕様を確認する。

○リトライ仕様

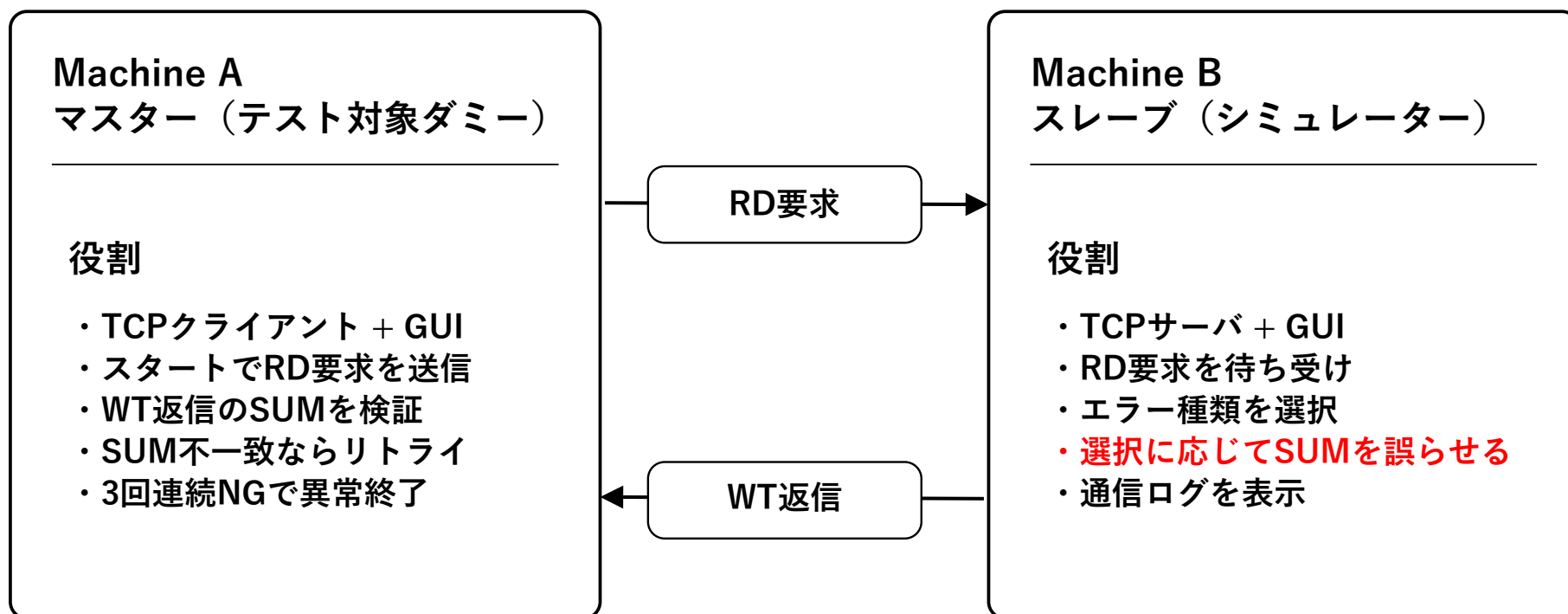
返信データに異常があった場合、マスターは最大2回までリトライする。それでも正常な返信が得られない場合は、異常終了とする。

○確認内容

- ✓ スレーブ側で返信伝文のSUM異常を発生させる
- ✓ エラー回数を 0回／1回／2回／3回 から選択する
- ✓ マスターが仕様どおりに再送・異常終了するかを確認する

エラー条件	期待動作
正常	1回目で正常終了
エラー1回	NG → OK
エラー2回	NG → NG → OK
エラー3回	NG → NG → NG → 異常終了

- ・ マスターとスレーブを別々のマシンで動かす実装
- ・ 両者は TCP/IP（既定ポート 5000）で通信を行う
- ・ 伝文は STX/ETX/SUM のバイト列
- ・ 2台構成で通信異常（SUM異常）を任意回数注入し、マスターのリトライ仕様を確認



● MASTER

テスト対象ダミー（マスター）

スタートでスレーブヘッダー要求を送信し、SUM 異常時は最大2回まで再送します。

スレーブIPアドレス

ポート

127.0.0.1

5000

クリア

スタート

通信ログ

● SLAVE

シミュレータ（スレーブ）

返すエラーの回数を選ぶと、マスターのデータ要求に対して SUM を誤らせて返信します。

エラーの種類



正常（エラー0回）

1回目から正常な SUM を返信



エラー1回

1回エラー後、正常を返信



エラー2回

2回エラー後、正常を返信



エラー3回

3回連続でエラーを返信

現在の設定：正常（エラー0回）

通信ログ

ログをクリア

サーバの待ち受けログがここに表示されます。

弊社が目指していること

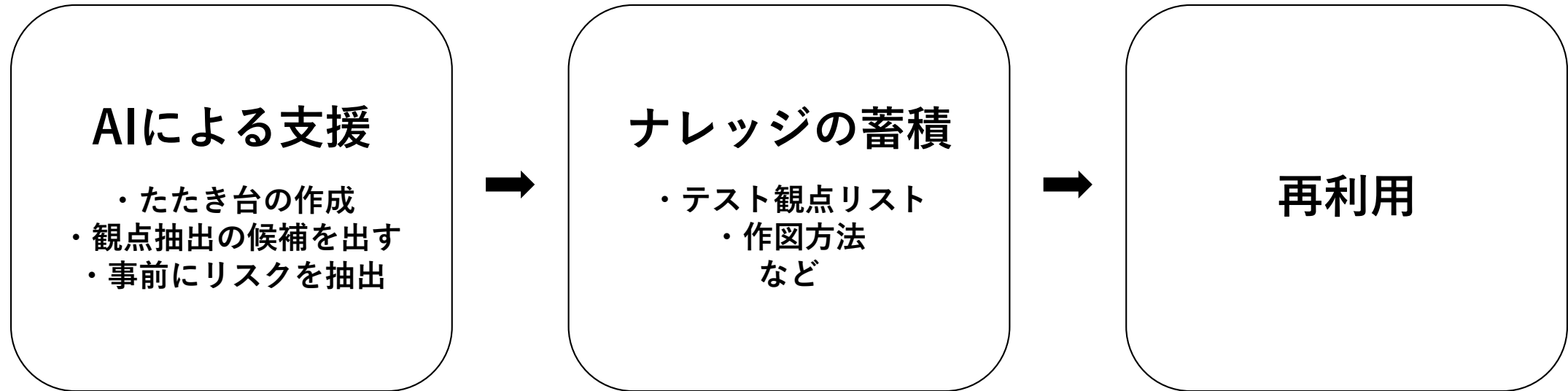
デモではシミュレーターを手動操作し確認



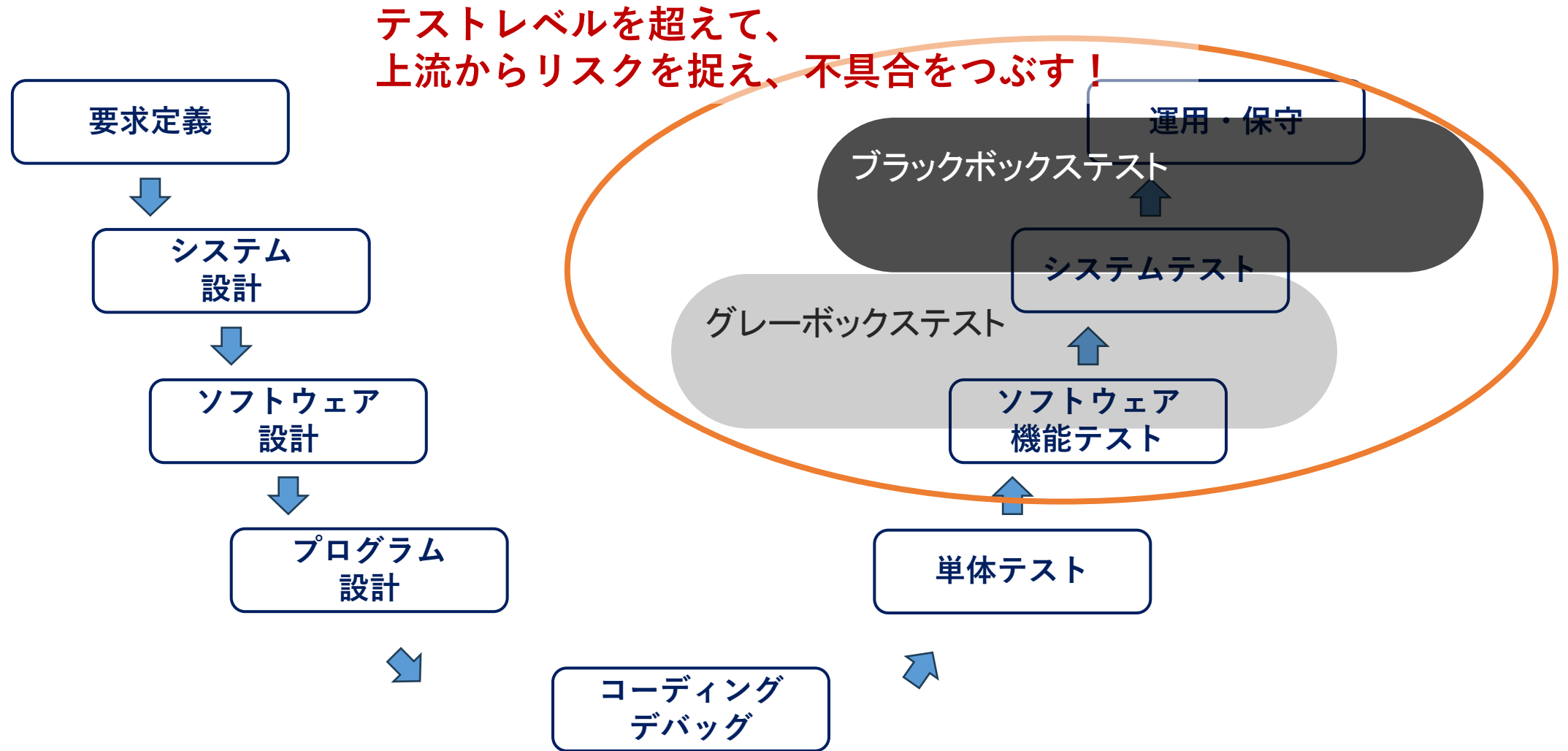
今後は自動化し、実機や周辺機器なしの状態でも
検証できる体制をつくる

1. GBでリスクを捉える
2. 異常条件をシナリオ化する
3. シミュレーターで自動注入する
4. 出力結果を確認する
5. 設計・実装へ早期にフィードバックする

目指すのは、上流で捉えたリスクを早く検証し、
不具合の芽を早い段階で潰せる状態をつくること



継続的な活動のために、成果物を資産として残し、設計・教育に再利用する。



✓BBは重要

ユーザー視点・仕様確認は、これからも重要。

✓GBで上流からリスクを捉える

内部状態・タイミング・連携のリスクを設計に取り込む。

仕様書をなぞるテストから、
仕組みを理解してリスクを捉えるテストへ



株式会社 iTest

お問い合わせはこちら

<https://www.itest.co.jp/contact>

